

Apellidos:

Nombre:

Matrícula:

---

# Examen

## 105000016 - Programación para Sistemas Grado en Ingeniería Informática

Lenguajes y Sistemas Informáticos e Ingeniería de Software

Facultad de Informática

Universidad Politécnica de Madrid

Curso 2012/2013 - Junio 2013

### Normas

- El examen puntúa sobre **12 puntos**.
- La duración total del mismo es de **una hora y cuarto**.
- Se deberá tener el DNI o el carnet de la UPM en lugar visible.
- No olvidar rellenar **apellidos, nombre y número de matrícula** en cada hoja.
- La solución al examen se proporcionará antes de la revisión.
- La fecha prevista de publicación de calificaciones es el **25 de junio**, y se realizará a través del Aula Virtual de la asignatura.
- La revisión del examen tendrá lugar el **27 de junio** a las 12:00 en la sala 2319.

### Cuestionario

(1 punto) 1. Suponiendo que las variables A y B contienen números enteros válidos. ¿Cuál de los siguientes mandatos comprueba si \$A es menor que \$B?

A. [ \$A -gt \$B ]    B. [ \$A -ge \$B ]    **C. [ \$A -lt \$B ]**    D. [ \$A -le \$B ]

(1 punto) 2. Dado el siguiente script `parametros.sh`:

```
#!/bin/bash  
echo $*
```

¿Cuál es el resultado de la siguiente invocación del mismo desde la línea de mandatos?

```
$ ./parametros.sh esto es una prueba
```

**Solución:**

esto es una prueba

- (1 punto) 3. Supóngase el resultado del siguiente mandato en una máquina Unix:

```
$ ls -al /etc/passwd /etc/shadow
-rw-r--r-- 1 root root    1970 mar 15 00:31 /etc/passwd
-rw-r----- 1 root shadow 1286 mar 15 00:31 /etc/shadow
```

¿Cuál sería la salida estándar del siguiente mandato Bash ejecutado por un usuario sin privilegios (no es root ni pertenece a grupos de administración como adm o shadow)?<sup>1</sup>

```
for f in passwd group; do
    echo -n "$f:_"
    test -r /etc/$f && echo -n "R"
    echo -n ";"
done
```

- A. passwd: R; shadow: R;
- B. passwd: R; shadow: ;**
- C. passwd: ; shadow: R;
- D. passwd: ; shadow: ;

- (1 punto) 4. En el manual de Bash, se puede leer la siguiente descripción sobre la expansión de variables<sup>2</sup>:

```
${var#palabra}
```

Elimina el prefijo. Si "palabra" encaja con el comienzo del valor de "var" entonces el resultado de la expansión es el valor de "var" eliminando el prefijo. Si no encaja el resultado es el valor de "var".

Escribir las tres líneas de la salida estándar resultado de la ejecución de los siguientes mandatos Bash:

```
unset X
X="0123456789"
echo ${X#0}
echo ${X#1}
echo ${X#012345}
```

**Solución:**

```
123456789
0123456789
6789
```

<sup>1</sup>`echo -n` no escribe el cambio de línea

<sup>2</sup>La construcción es en realidad mucho más compleja pero estas líneas bastan para resolver el ejercicio.

- (1 punto) 5. Dado el siguiente código C, que mostraría por la salida estándar

```
#include<stdio.h>
int main()
{
    char cadena[10] = {'H',' ','o',' ','l',' ','a'};
    printf("%s",cadena);
}
```

- A. Hola  
B. Hola más exactamente seis caracteres imprevisibles.  
**C. Hola más otros caracteres imprevisibles.**

- (1 punto) 6. Indique las sentencias necesarias en un programa C para abrir el fichero /home/mio/leer.txt en modo lectura. Y, además, imprima en la salida estándar "Abierto" si se ha podido abrir.

**Solución:**

```
#include<stdio.h>
int main()
{
    FILE *ppa;
    if (paa = fopen("/home/mio/leer.txt","r")) {
        printf("Abierto\n");
    }
    return 0;
}
```

- (2 puntos) 7. En un programa C tenemos una función que cada vez que se la llama devuelve un número:

```
int devuelve(void);
```

Escribir el código necesario para llamar repetidamente a dicha función hasta que devuelva un número estrictamente mayor que cero, guardando este último valor (el positivo) en una variable que se llame x.

**Solución:**

```
#include<stdio.h>
int main()
{
    int x;
    do {x = devuelve();} while (x <= 0);
}
```

- (1 punto) 8. Escriba la orden para, a partir del fichero `programa.c`, construir el fichero objeto, `programa.o`, sin construir el ejecutable.

**Solución:**

```
$ gcc -c programa.c -o programa.o
```

o bien

```
$ gcc -c programa.c
```

- (1 punto) 9. ¿Qué tamaño tiene en `triqui.fi.upm.es` una variable de tipo **long**?  
¿Y una de tipo **char**?

**Solución:** 8 bytes y un byte respectivamente.

- (2 puntos) 10. Definir un **struct** que permita representar los nodos de cadenas simplemente enlazadas (típicas en la implementación de pilas, colas, etc.).

**Solución:**

```
struct nodo_s {  
    int elemento;  
    struct nodo_s *siguiente;  
};
```